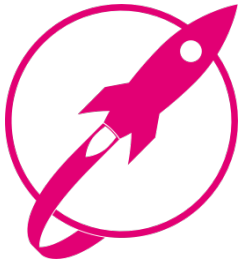


FH·W-S

iOS Programmierung

(mit Swift)

Peter Braun, Florian Bachmann & Andreas Wittmann
@pe_braun @florianbachmann @anwittmann

Deutsche Telekom AG
FHWS - Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt

#FHWSSwift

Agenda

1. **Introduction** – Organisatorisches
2. **First iOS-Project** – Hello World, **First iOS-Project** – Still Hello World (now with Code 😄)
3. **Swift**, Wait!, What about Objective-C?, Why Swift?
4. **A (not so) Quick Tour**
5. **Documentation**
6. **The basics –** iOS Architecture & more
7. **User Interfaces –** View Controller, Auto Layout & Size Classes
8. **Storyboard & Segues**
9. **Tables & NavigationController**
10. **TabBarController**
11. **Notifications**
12. **PickerViews**
13. **Touches, Gestures, 3D Touch, Peek & Pop**
14. **ScrollView & StackViews**
15. **Networking** – JSON & Dependency Managers
16. **WebKit**
17. **Maps**
18. **Storage & Data persistency –** NSUserDefaults, NSKeyedArchiver & Core Data
19. **ObjC**

Touch
Touches, Gestures & 3D

Touches (1/2)

We differentiate between two kind of events
Touch and Motion, we will focusing on Touch:

```
func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {}

func touchesMoved(touches: Set<UITouch>, withEvent event: UIEvent?) {}

func touchesEnded(touches: Set<UITouch>, withEvent event: UIEvent?) {}

func touchesCancelled(touches: Set<UITouch>, withEvent event: UIEvent?) {}

//Report changes in the amount of force associated with a touch event
func touchesEstimatedPropertiesUpdated(touches: Set<UITouch>){} - iOS 9.1
```

Touches (2/2)

Because iOS devices are multitouch devices, the touches method return an set of touches (e.g. multi touch points).

```
func touchesBegan(touches: Set<UITouch>, withEvent event: UIEvent?) {
    let touch = touches.first!
    start = touch.locationInView(view)
}

func touchesMoved(touches: Set<UITouch>, withEvent event: UIEvent?) {
    let touch = touches.first!
    let end = touch.locationInView(view)
    …
}
```

example how to get only the first touch

please read API Documentation of locationInView

Touch in iOS 9.1

In iOS 9 the touch events get an update, two new Properties are available

- force - CGFloat:

```
The force of the touch
```

- type - Enum:

```
Direct

iPhohne/ iPad…

finger touch
```

```
Indirect

Apple TV

Apple TV Remote
```

```
Stylus

Apple Pencil
```

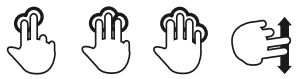
TouchPainter
Demo

TouchPainter - Demo

```
func drawFromPoint(canvas:UIImageView, start: CGPoint, toPoint end: CGPoint ,
withColor:UIColor, lineWidth:CGFloat, alpha:CGFloat) -> UIImage {
    UIGraphicsBeginImageContext(canvas.frame.size)
    let context = UIGraphicsGetCurrentContext()
    canvas.image?.drawInRect(CGRect(x: 0, y: 0,
        width: canvas.frame.size.width, height: canvas.frame.size.height))
    CGContextSetLineWidth(context, lineWidth)
    CGContextSetAlpha(context, alpha);
    CGContextSetStrokeColorWithColor(context, withColor.CGColor)
    CGContextBeginPath(context)
    CGContextMoveToPoint(context, start.x, start.y)
    CGContextAddLineToPoint(context, end.x, end.y)
    CGContextStrokePath(context)
    let newImage = UIGraphicsGetImageFromCurrentImageContext()
    UIGraphicsEndImageContext()
    return newImage
}
```

Gestures
Touches, Gestures & 3D

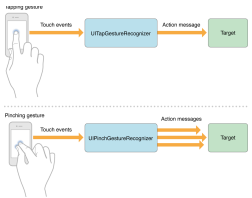
Standard Gestures

View	Purpose
 Tap	To press or select a control (or item)
 Double tap	To zoom in/out and center a block of content or an image
 Touch & hold	In editable text, to change the cursor position
 Swipe/Pan	To scroll or pan (In a table-view row, to reveal the additional buttons)
 Flick	To Scroll or pan quickly
 Pinch open	Zoom in
 Pinch close	Zoom out
	many more

Source: Apple - Start Developing iOS Apps Today

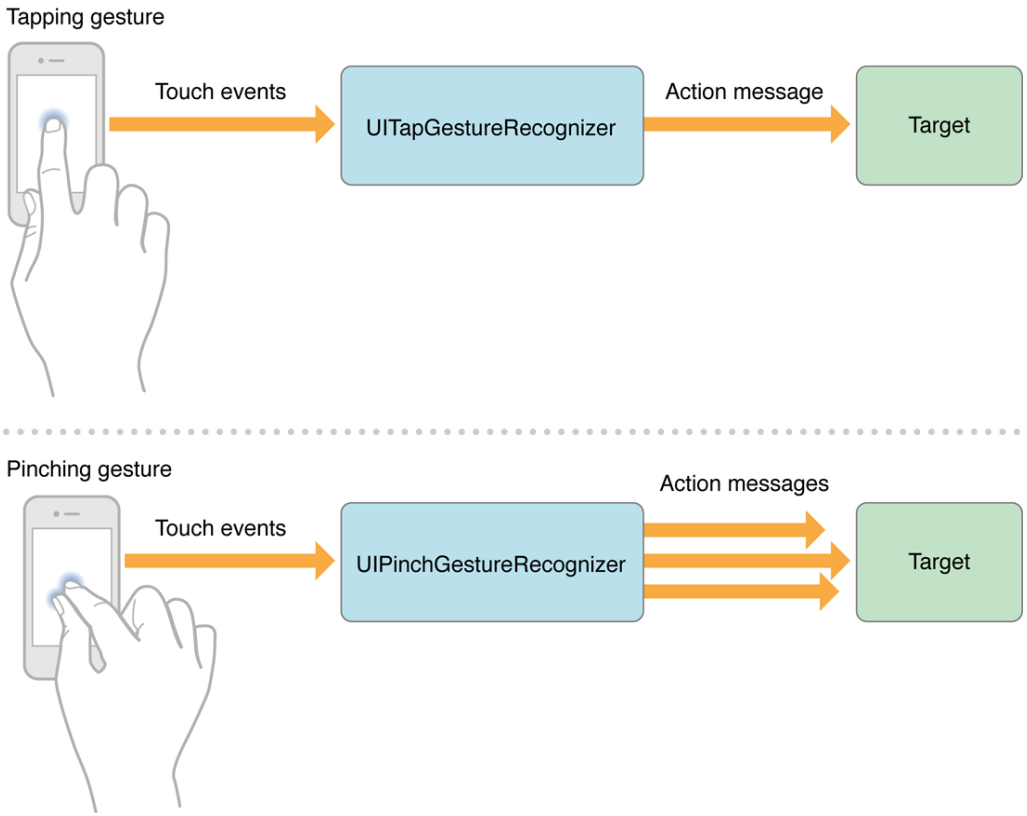
Gesture Recognizers (1/3)

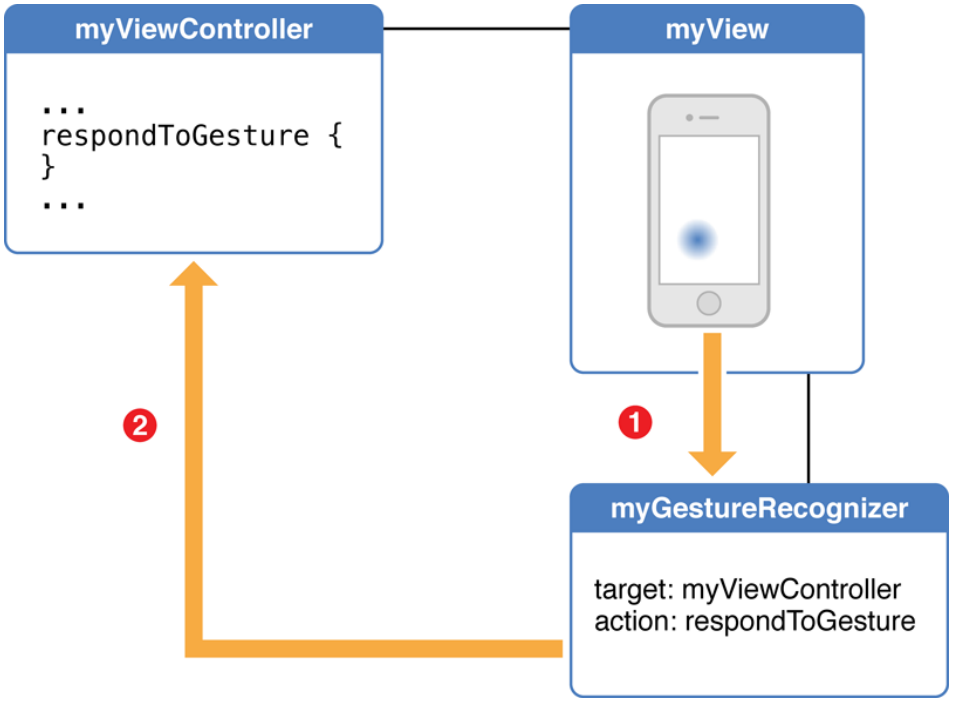
UIKit class	Gesture
`UITapGestureRecognizer`	Tapping (any number of taps)
`UIPinchGestureRecognizer`	Pinching in and out (for zooming a view)
`UIPanGestureRecognizer`	Panning or dragging
`UISwipeGestureRecognizer`	Swiping (in any direction)
`UIRotationGestureRecognizer`	Rotating (fingers moving in opposite directions)
`UILongPressGestureRecognizer`	Long press (also known as “touch and hold”)



Source: Event Handling Guide for iOS

Gesture Recognizers (2/3)





Source: Event Handling Guide for iOS

Gesture Recognizers (3/3)

```
...

var tapGesture = UITapGestureRecognizer(target: self, action: “SomeMethod:")
tapGesture.numberOfTapsRequired = 1;
self.view.addGestureRecognizer(tapGesture)

...

func SomeMethod(recognizer : UITapGestureRecognizer) {
    println("tap detected")
}
```

Source: Event Handling Guide for iOS

GestureTutorial

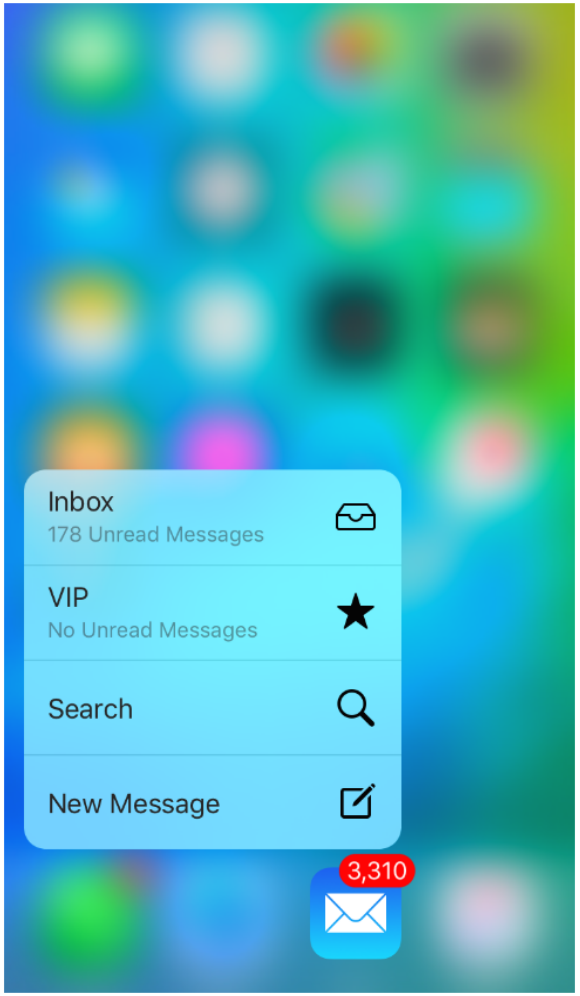
Demo

Tipps:

- Auf einem iPad/iPhone funktionieren Gesten besser als im Simulator (Peter gibt euch gerne Test-Geräte)
- CGAffineTransformScale(sender.view!.transform, sender.scale, sender.scale)
- CGAffineTransformRotate(currentTrans, rotation)
- sender.translationInView(self.view);sender.view!.center = CGPoint...;
- sender.setTranslation(...)`

3D
Touches, Gestures & 3D

Quick Actions (1/9)



Long Press on an icon,
opens a direct access
to the main functionalities of an app

Quick Actions (2/9)

http://www.apple.com/iphone-6s/3d-touch/

Quick Actions (3/9)

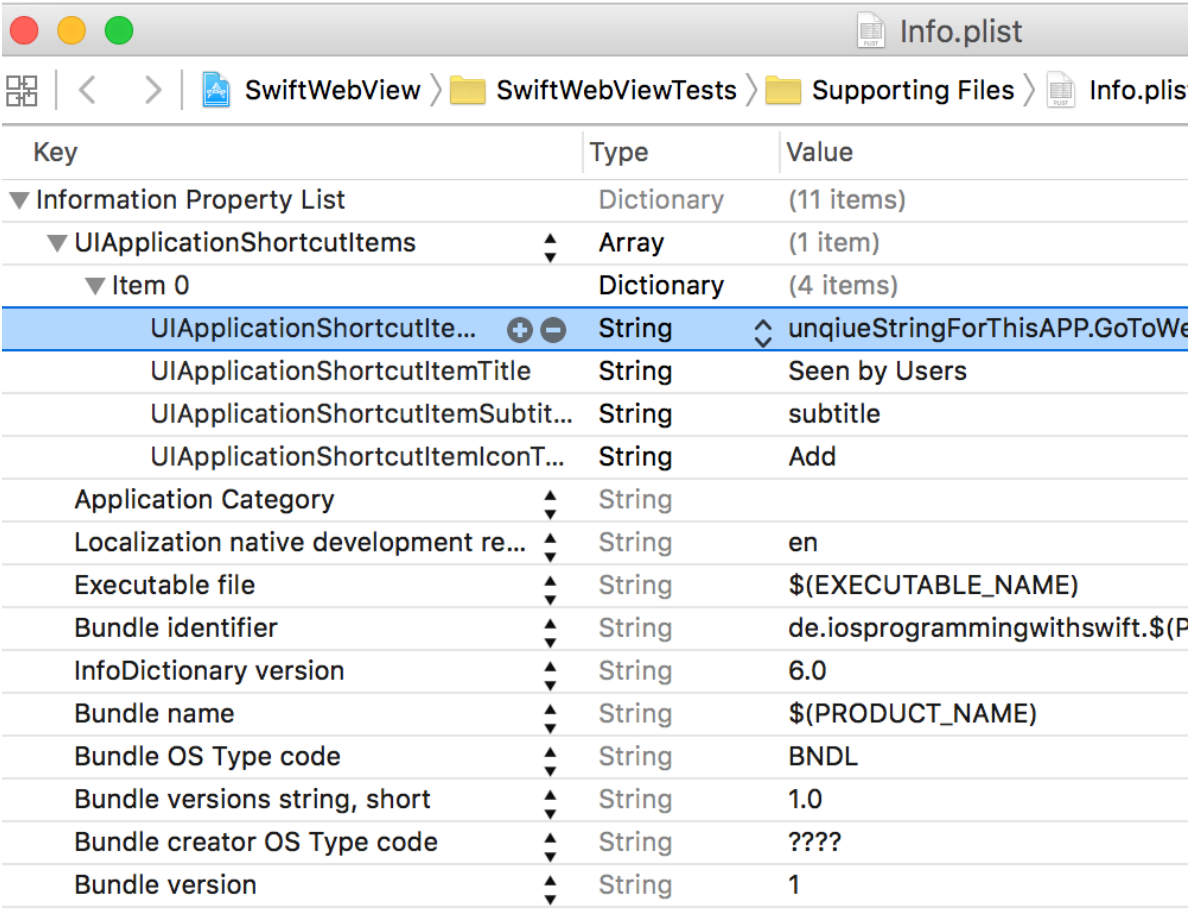

```
Style is defined in the
.plist of an app
```

Quick Actions (4/9)

Key	Type	Value
▼ Information Property List	Dictionary	(11 items)
▼ UIApplicationShortcutItems	Array	(0 items)
Application Category	String	
Localization native development re...	String	en
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	de.iosprogrammingwithswift.$(PRODUCT_N
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	BNDL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1

```
In .plist:
generate:
UIApplicationShortcutItems as
Array
```

Quick Actions (5/9)



In UIApplicationShortcutItems:

add a Row as Dictionary

Quick Actions (6/9)

Parameters for the dictionary
blue is required

Parameters	Description
UIApplicationShortcutItemType	Unique string to identify the action
UIApplicationShortcutItemTitle	Title for the Item of an action
UIApplicationShortcutItemSubtitle	Subtitle for the Item
UIApplicationShortcutItemIconType	Icon for the Item (System Icons)
UIApplicationShortcutItemIconFile	Icon name of the image file in the bundle or assets catalog. (Template)
UIApplicationShortcutItemUserInfo	dictionary with custom data you want to use.

Quick Actions (7/9)

in AppDelegate.swift

```
@available(iOS 9.0, *)
func application(
    application: UIApplication,
    performActionForShortcutItem shortcutItem: UIApplicationShortcutItem,

    completionHandler: (Bool) -> Void) { print("Shortcut tapped”) }

)
```

Quick Actions (8/9)

in AppDelegate.swift add some more code to handle different actions:

```
 @available(iOS 9.0, *)
func application(
     application: UIApplication,
     performActionForShortcutItem shortcutItem: UIApplicationShortcutItem,

     completionHandler(handleShortcut(shortcutItem))

)
```

```
func handleShortcut( shortcutItem: UIApplicationShortcutItem ) -> Bool {
        switch shortcutItem.type{
            case "unqiueStringForThisAPP.GoToWebSite":
                return true;
        default:
                return false;
        }
    }
```

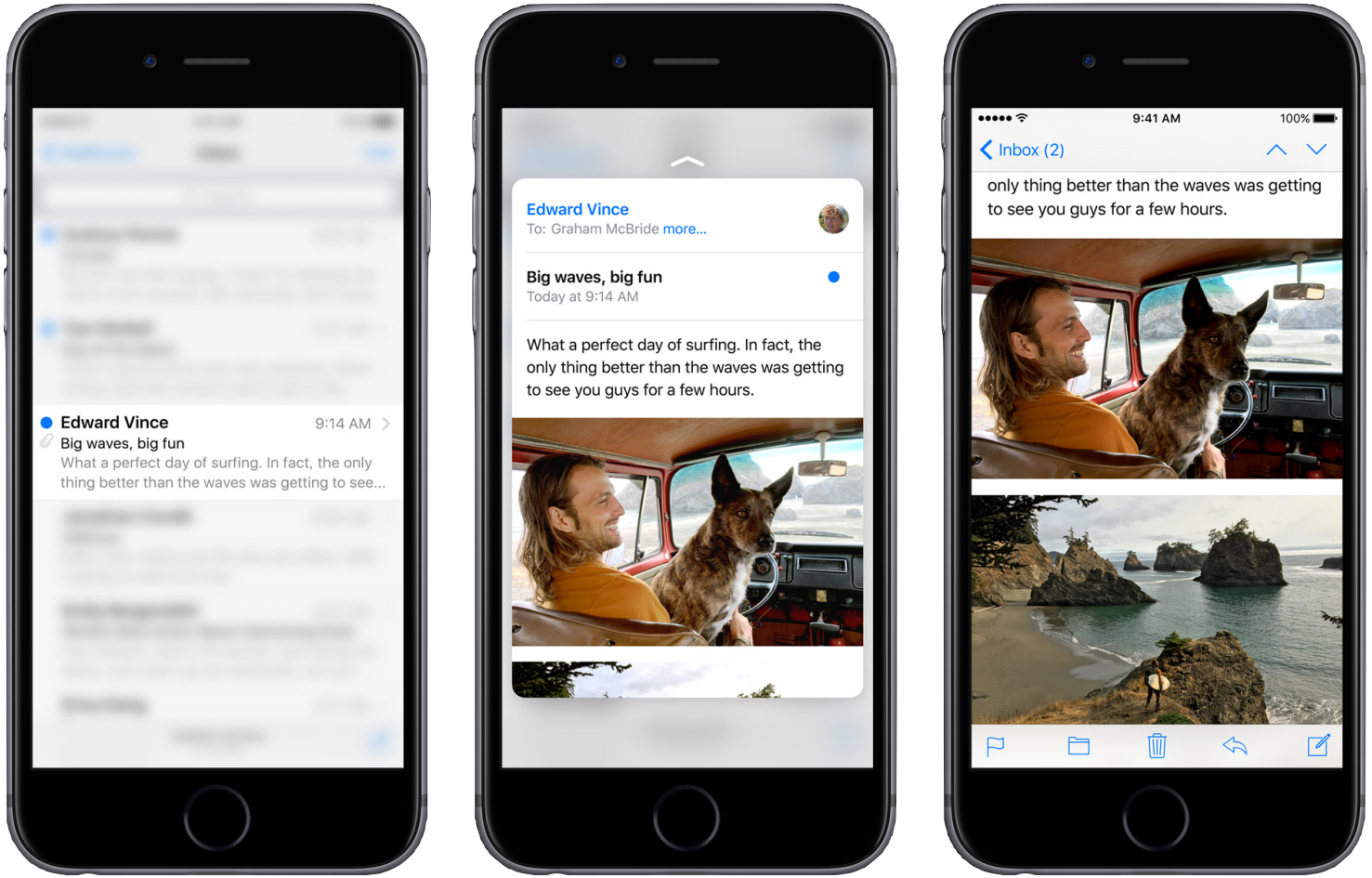
Quick Actions (9/9)

To avoid side effects we check in the didFinishLaunchingWithOptions function, if the app is started over an shortcut Item or an click on the icon.

```
func application(application: UIApplication,
                 didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?)
     -> Bool
    {
        if let shortCutItem = launchOptions?
                    [UIApplicationLaunchOptionsShortcutItemKey]
                    as? UIApplicationShortcutItem {
                handleShortcut(shortCutItem)
            return false
        }
        return true
    }
```

Peek & Pop

Peek & Pop (1/5)

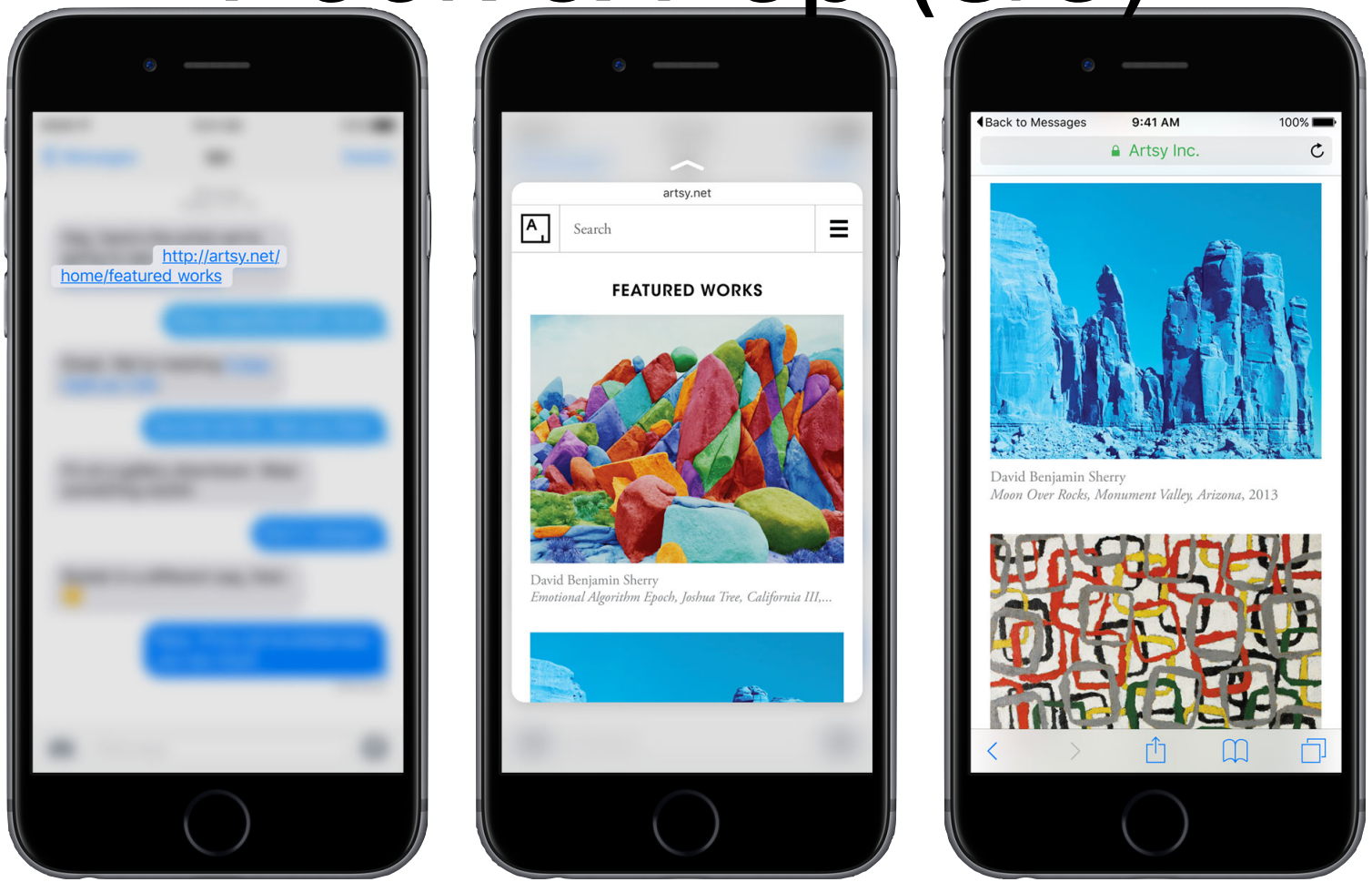


http://www.apple.com/iphone-6s/3d-touch/

Peek & Pop (2/5)

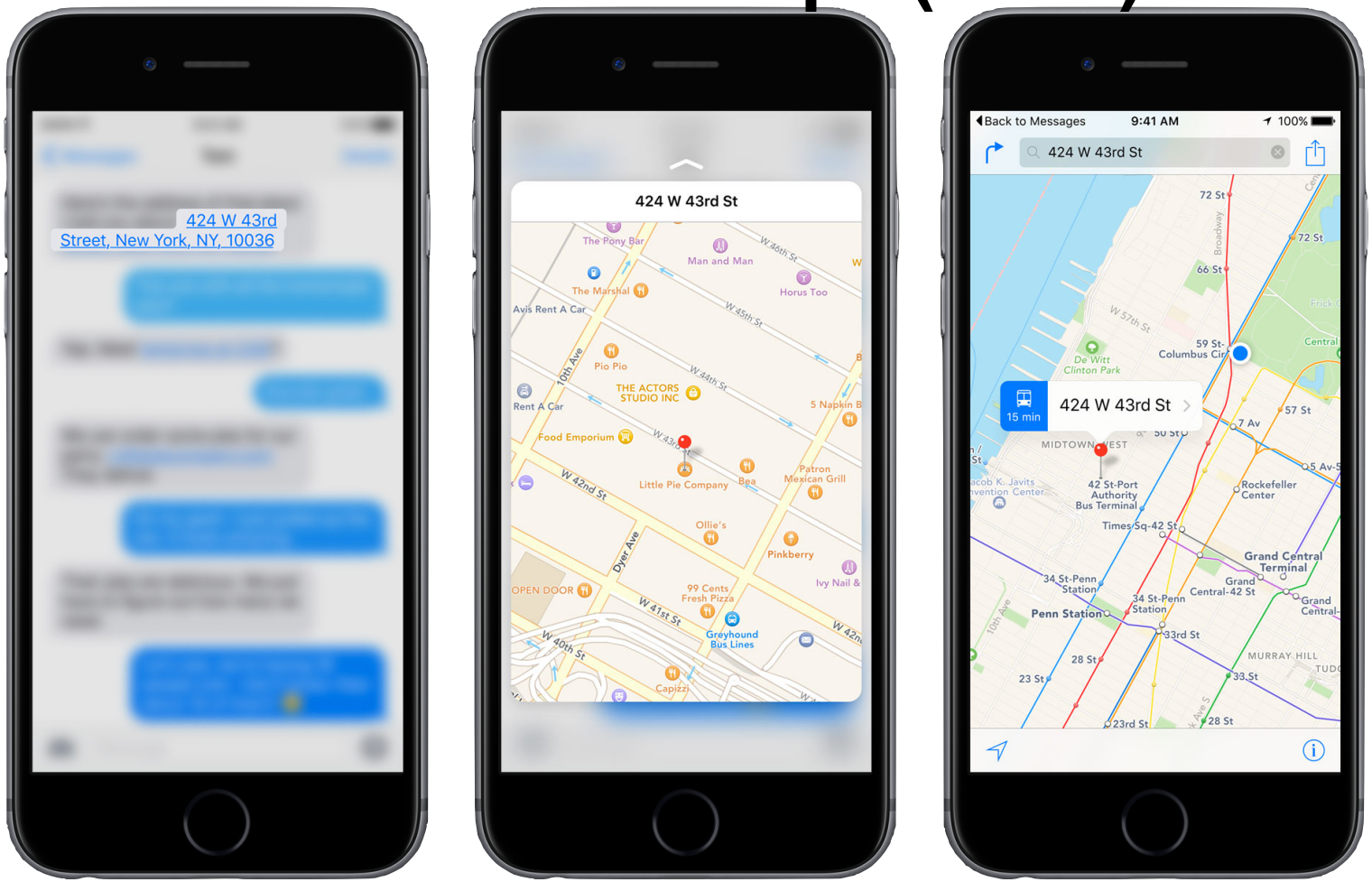
http://www.apple.com/iphone-6s/3d-touch/

Peek & Pop (3/5)



http://www.apple.com/iphone-6s/3d-touch/

Peek & Pop (4/5)

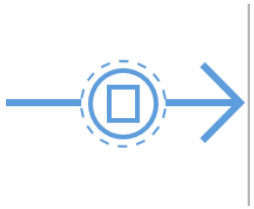


http://www.apple.com/iphone-6s/3d-touch/

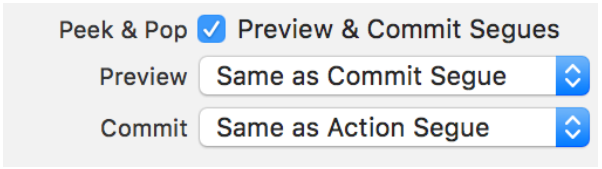
Peek & Pop (5/5)

In IB select a segue and enable Peek & Pop in Utilities.

(1)



(2)



Peek & Pop
Quick Demo